A COMPUTER ROUTINE FOR THE ESTIMATION OF VARIANCE COMPONENTS IN THE GENERAL MIXED MODEL BY THE RESTRICTED MAXIMUM LIKELIHOOD (REML) PROCEDURE

by

BU-543-M January, 1975

R. R. Corbeil

Abstract

This documentation supports a program which computes estimates of fixed effects and variance components in the general mixed model by the REML procedure described in Corbeil and Searle (1974). The program language is Fortran IV for the IBM systems 360 and 370 series. A listing of the source code is appended to the documentation.

Introduction

The theory of restricted maximum likelihood (REML) estimation of variance components in the general mixed model and the algorithm which forms the basis of the computer routine is fully described in Corbeil and Searle (1974). The model for $\underset{\sim}{y}$, a vector of N observations, is

$$\underset{\sim}{y} = \underset{\sim}{X}\underset{\sim}{\mu} + \underset{\sim}{U}_1\underset{\sim}{b}_1 + \cdots + \underset{\sim}{U}_c\underset{\sim}{b}_c + \underset{\sim}{e}$$

and

$$\underset{\sim}{y} \sim N\left(\underset{\sim}{X}\underset{\sim}{\mu},\ \sum_{i=1}^{c} \sigma_i^2 \underset{\sim}{U}_i\underset{\sim}{U}_i' + \underset{\sim}{I}_N\sigma^2\right),$$

where

$\underset{\sim}{\mu}$ is a vector of k unknown constants, the fixed effects of the model,

$\underset{\sim}{X}$ is an $N \times k$ incidence matrix, of full column rank, corresponding to $\underset{\sim}{\mu}$ and with $k < N$,

Paper No. BU-543-M in the Biometrics Unit Mimeo Series, Department of Plant Breeding and Biometry, Cornell University, Ithaca, New York 14853.

$\underset{\sim}{U}_i$ is an $N \times m_i$ design matrix associated with the i^{th} random factor, with $\sum_{i=1}^{c} m_i + k < N$,

$\underset{\sim}{b}_i$ is a vector of m_i random variables which are i.i.d. $N(0, \sigma_i^2)$, with the $\underset{\sim}{b}_i$'s being mutually independent,

and

$\underset{\sim}{e}$ is a vector of N random variables which are i.i.d. $N(0, \sigma^2)$ and independent of the $\underset{\sim}{b}_i$'s.

As part of the necessary information to be provided to the program is the $\underset{\sim}{y}$ vector of N observations, the matrices $\underset{\sim}{U}_1, \cdots, \underset{\sim}{U}_c$, $\underset{\sim}{X}$ and the constants N, k, c, and all m_i's. The program will compute by an iterative procedure the estimates of c variance components in the form of $\gamma_i = \sigma_i^2/\sigma^2$, σ^2, an objective function (which is being minimized) and the c first order partials. These are printed at each iteration. Upon the simultaneous convergence of the c γ-components, estimates of $\underset{\sim}{\mu}$ are computed. These are also printed along with the γ_i's, the σ_i^2's, and their estimated variances and covariances.

Description of input card deck

The user card deck consists of a title card, 2 or more control cards containing the constants of the model and several program control parameters, 2 format control cards, and the data.

1. Title card (1^{st} card)

   This is a single card which contains a message in columns 1-80 that the user may wish to have appear as a heading to his output.

2. Control cards

   (a) The 2^{nd} card contains the following constants in the format 7I5:

N = the total number of observations

K = k, the number of fixed effects in the model

IC = c, the number of random effects in the model

M = $\sum_{i=1}^{c} m_i$, the total number of levels of all random factors

IT = the maximum number of iterations to be allowed should the stopping rule fail

IOUT = $\begin{cases} \text{0 or blank, if the input data } (\underset{\sim}{U}_1 \cdots \underset{\sim}{U}_c \underset{\sim}{X} \underset{\sim}{y}) \text{ is not to be listed} \\ \text{some integer value, if the input data is to be listed} \end{cases}$

IPREC = the number of significant digits required of the estimation procedure, the default value is 10

(b) The 3rd card and possibly one or more subsequent cards contain the c m_i values of the model in the format 16I5, where m_i is the number of levels of the i^{th} random factor, i.e., the number of columns in $\underset{\sim}{U}_i$. The m_i's should be listed in numerical sequence for i = 1, ···, c to a maximum of 16 per card.

3. Format cards

There are two format cards which follow in sequence after the control cards described above:

(a) The data which is taken to be in the form of an augmented matrix $(\underset{\sim}{U}_1 \cdots \underset{\sim}{U}_c \underset{\sim}{X} \underset{\sim}{y})$ is read one row at a time and a format card, e.g., (6F3.0,F5.1), is necessary for locating each of its elements.

(b) Another format card, e.g. (1X,5F3.0,3X,F3.0,3X,F5.1), controls the listing of the augmented matrix if the IOUT option is > 0. If IOUT = 0 or blank, a blank card or dummy format card is necessary to assure the proper sequence of cards is read by the program.

4. <u>Data cards</u>

(a) There are N sets of data cards to be read under the control of format card (3a) described above. Each set represents a row of the $(\underset{\sim}{U}_1 \cdots \underset{\sim}{U}_c \underset{\sim}{X} \underset{\sim}{y})$ matrix.

(b) The iterative procedure requires initial estimates of the γ_i's. These are read in under the format 8F10.0. It is desirable to use the very best estimates available, but should none be available, setting each $\gamma_i = 1.0$ is usually adequate. NOTE: If the initial estimates of the γ_i's are such that the objective function increases for 2 successive iterations, processing is discontinued.

Program description

The program is written in 4 parts: a mainline which contains the dimension statement, a subroutine which carries out the bulk of the calculations, and two inversion subroutines, each being applied to a special task. Although a source listing is attached, the strategy which is prescribed by these subroutines is detailed in the appendix of Corbeil and Searle (1974).

The minimum dimensions to be specified in the mainline are:

MI (c + 1),

R (m + k + 1, m + k + 1), where $m = \sum_{i=1}^{c} m_i$,

W (m + k + 1, m + k + 1),

D (c),

WM (f, f), where f = max(m, k + 1),

WM2 (c, f),

DR (c),

and TITLE (20),

with c and k as previously defined.

```
//XF981    JOB XXXXXXX,'810--R. R. CORBEIL  ',PRTY=6                  DLVR=WRN
/*LIMITS REGION=100K,CLASS=C,CPUTIME=(,20)
// EXEC FORTGCLG
C*****RESTRICTED MAXIMUM LIKELIHOOD (REML) ESTIMATION OF VARIANCE
C*****COMPONENTS FOR THE MIXED MODEL (BU-538-M) BASED ON THE HEMMERLE
C*****AND HARTLEY ALGORITHM (TECHNOMETRICS 15:819-831, 1973).  THE
C*****DOCUMENTATION FOR THIS PROGRAM IS (BU-543-M), A BIOMETRICS UNIT,
C*****CORNELL UNIVERSITY, ITHACA, N. Y. MIMEOGRAPH SERIES.
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION MI(2),R(10,10),W(10,10),D(1),WM(7,7),WM2(1,7),
     1DR(1),TITLE(20)
  120 READ(5,102,END=999) (TITLE(I),I=1,20)
  102 FORMAT(20A4)
      WRITE(6,103) (TITLE(I),I=1,20)
  103 FORMAT('1','**********  REML:  ',20A4,'  **********')
      READ(5,110)N,K,IC,M,IT,IOUT,IPREC
  110 FORMAT(7I5)
      WRITE(6,111)N,K,IC,M,IT
  111 FORMAT('0N=',I5,'   K=',I5,'   C=',I5,'   M=',I5,'   IT=',I5)
      MI(1)=0
      K1=K+1
      K2=M+K1
      IC1=IC+1
      READ(5,112)(MI(I+1),I=1,IC)
  112 FORMAT(16I5)
      WRITE(6,113)(I,MI(I+1),I=1,IC)
  113 FORMAT(16I5)
      IF(M.GT.K1)GO TO 100
      MAXMK1=K1
      GO TO 101
  100 MAXMK1=M
  101 CALL PMAIN(MI,R,W,D,DR,WM,WM2,N,K,IC,M,IT,K1,K2,IC1,MAXMK1,
     +IOUT,IPREC)
      GO TO 120
  999 STOP
      END

      SUBROUTINE PMAIN(MI,R,W,D,DR,WM,WM2,N,K,IC,M,IT,K1,K2,IC1,MAXMK1,
     +IOUT,IPREC)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION MI(IC1),R(K2,K2),W(K2,K2),D(IC),DR(IC),
     & WM(MAXMK1,MAXMK1),WM2(IC,MAXMK1)
      DIMENSION FMT1(18),FMT2(18)
      READ(5,200)FMT1
      READ(5,200)FMT2
  200 FORMAT(18A4)
      TOL=.5D-10
      EPS=.001D0
      PREC=1.0D1**(-IPREC)
      IF(IPREC.EQ.0) PREC=1.0D-10
      IT1=0
      IEND=0
      TOBF=0.0D0
      M1=M+1
      K4=K2
C      INPUT (V X Y) AND FORM R
      DO 1 I=1,K2
      DO 1 J=I,K2
    1 R(I,J)=0.
      WRITE(6,202) IOUT
  202 FORMAT('0OUTPUT DATA OPTION = ',I5)
      DO 2 L=1,N
      READ (5,FMT1)  (W(1,I),I=1,K2)
```

```
      IF (IOUT.GT.0) WRITE(6,FMT2) (W(1,I),I=1,K2)
      DO 2 I=1,K2
      DO 2 J=I,K2
2     R(I,J)=R(I,J)+W(1,I)*W(1,J)
C        REARRANGE SUBMATRICES OF R TO COINCIDE WITH ( V Y X )
      DO 59 I=1,K2
59    W(2,I)=R(I,K2)
      DO 64 I=1,M
      DO 64 J=1,K
64    R(I,K2-J+1)=R(I,K2-J)
      DO 68 I=1,K
      DO 68 J=1,I
68    R(K2-I+1,K2-J+1)=R(K2-I,K2-J)
      R(M1,M1)=W(2,K2)
      DO 69 I=1,M
69    R(I,M1)=W(2,I)
      DO 84 I=1,K
84    R(M1,M1+I)=W(2,M+I)
      DO 3 I=1,K2
      DO 3 J=I,K2
3     R(J,I)=R(I,J)
C      INPUT INITIAL VALUES OF D
29    READ(5,203)(D(I),I=1,IC)
203   FORMAT(8F10.0)
      DO 44 I=1,IC
44    DR(I)=DSQRT(D(I))
C        INVERT X'X PORTION OF R
      M2=M+2
      CALL INV2(R,K,0,M2,K2,DD,K2,M1)
C        TRANSFORM R   ( M1 X M1 )
      DO 31 I=1,M
      DO 31 J=1,K
      W(I,J)=0.0D0
      DO 31 II=1,K
31    W(I,J)=W(I,J)+R(I,M1+II)*R(M1+II,M1+J)
      DO 32 I=1,M
      DO 32 J=1,M
      DO 32 II=1,K
32    R(I,J)=R(I,J)-W(I,II)*R(M1+II,J)
      DO 24 I=1,M
      DO 33 II=1,K
33    R(I,M1)=R(I,M1)-W(I,II)*R(M1+II,M1)
24    R(M1,I)=R(I,M1)
      YY=0.0D0
      DO 34 I=1,K
      W(1,I)=0.0D0
      DO 35 II=1,K
35    W(1,I)=W(1,I)+R(M1,M1+II)*R(M1+II,M1+I)
34    YY=YY+W(1,I)*R(M1+I,M1)
      R(M1,M1)=R(M1,M1)-YY
      K3=K2
      NK=N-K
      WRITE(6,209)
209   FORMAT('-'/'0','ITERATION NO.',8X,'OBJECTIVE FUNCTION',8X,'SIGMA S
     &QUARED',12X,'FIRST ORDER PARTIALS',16X,'GAMMA')
      GO TO 30
6     CONTINUE
C      COMPUTE SIGMA SQUARED
      SIGMA=W(K2,K2)
      SIGMA=SIGMA/NK
C      COMPUTE OBJECTIVE FUNCTION (MODIFIED LIKELIHOOD)
      OBF=(N-K)*DLOG(SIGMA)+DET
      IF (OBF.LT.TOBF) GO TO 401
      IEND=IEND+1
      GO TO 402
401   IEND=0
```

```
402    TOBF=OBF
C       COMPUTE 1ST ORDER PARTIALS (NEGATIVES)
       LL=1
       LU=0
       DO 11 I=1,IC
       LL=LL+MI(I)
       LU=LU+MI(I+1)
       WM(I,IC1)=0
       L1=1
       DO 12 L=LL,LU
       WM(I,IC1)=WM(I,IC1)+W(L,L)
       WM2(I,L1)=W(K2,L)
12     L1=L1+1
       TEMP=0
       L1=1
       DO 14 L=LL,LU
       TEMP=TEMP+WM2(I,L1)*WM2(I,L1)
14     L1=L1+1
       TEMP=TEMP/(2.0D0*SIGMA)
       WM(I,IC1)=(WM(I,IC1)/2.D0)-TEMP
11     WM(I,IC1)=DR(I)*WM(I,IC1)
C       INTERMEDIATE OUTPUT
       WRITE(6,210) IT1,OBF,SIGMA,WM(1,IC1),D(1)
210    FORMAT('0',10X,I2,11X,G18.10,3X,G18.10,8X,'  1 ',2X,G18.10,9X,
      & G18.10)
       IF (IC.EQ.1) GO TO 13
       DO 48 I=2,IC
48     WRITE(6,211) I,WM(I,IC1),D(I)
211    FORMAT(' ',70X,I3,3X,G18.10,9X,G18.10)
13     IPC=0
       IF (IEND.GE.3) GO TO 403
       DO 9 I=1,IC
9      IF(DABS(WM(I,IC1)).LE.PREC    )IPC=IPC+1
       IF(IPC.EQ.IC.OR.IT1.EQ.IT)GO TO 23
       IT1=IT1+1
C       COMPUTE SECOND ORDER PARTIALS
       LLI=1
       LUI=0
       DO 15 I=1,IC
       LLJ=LLI
       LUJ=LUI
       LLI=LLI+MI(I)
       LUI=LUI+MI(I+1)
       DO 16 J=I,IC
       LLJ=LLJ+MI(J)
       LUJ=LUJ+MI(J+1)
       TEMP=0.
       WM(I,J)=0.
       I2=1
       DO 17 I1=LLI,LUI
       J2=1
       DO 18 J1=LLJ,LUJ
       WM(I,J)=WM(I,J)+W(I1,J1)*W(I1,J1)
       TEMP=TEMP+WM2(I,I2)*W(I1,J1)*WM2(J,J2)
18     J2=J2+1
17     I2=I2+1
       TEMP=TEMP/SIGMA
       WM(I,J)=(WM(I,J)/2.0D0)-TEMP
16     WM(I,J)=2.D0*DR(I)*DR(J)*WM(I,J)
       WM(I,I)=WM(I,I)-WM(I,IC1)/DR(I)
       DO 15 J=1,I
15     WM(I,J)=WM(J,I)
C       SOLVE NEWTON-RAPHSON EQUATIONS FOR DR
C       (ZERO ORDER APPROXIMATION INCLUDED)
       DO 60 I=1,IC
       WM2(I,1)=WM(I,IC1)
```

```
60    WM2(I,2)=WM(I,I)
      DO 61 I=1,IC
      IF(DR(I).LT.EPS)GO TO 61
      CALL INV2(WM,IC,1,I,I,DD,MAXMK1,0)
61    CONTINUE
      DO 62 I=1,IC
      IF(DR(I).LT.EPS)GO TO 63
      DR(I)=DABS(DR(I)+WM(I,IC1))
      GO TO 62
63    DR(I)=DABS(WM2(I,1)/WM2(I,2))
      IF(DR(I).GT.EPS)GO TO 62
      DR(I)=0.
62    D(I)=DR(I)*DR(I)
C      REMOVE ZERO COMPONENTS FROM MODEL
55    DO 54 LOOP=1,IC
      IF(DR(LOOP).GT.0.)GO TO 54
      IF(IC.EQ.1)GO TO 56
      IEPS=LOOP
      WRITE(6,205)IEPS
205   FORMAT('0COMPONENT ',I4,'  IS ZERO AND WILL BE DELETED')
      IX=MI(IEPS+1)
      I1=0
      DO 50 J=1,IEPS
50    I1=I1+MI(J)
      I2=I1+IX
      L=K3-I2
      DO 51 I=1,L
      DO 51 J=1,L
51    R(I1+I,I1+J)=R(I2+I,I2+J)
      IF(IEPS.EQ.1)GO TO 58
      DO 57 I=1,I1
      DO 57 J=1,L
      R(I,I1+J)=R(I,I2+J)
57    R(I1+J,I)=R(I2+J,I)
58    ICT=IC-1
      M=M-IX
      K3=K3-IX
      IF(IEPS.EQ.IC)GO TO 53
      DO 52 I=IEPS,ICT
      DR(I)=DR(I+1)
      D(I)=D(I+1)
52    MI(I+1)=MI(I+2)
      GO TO 53
54    CONTINUE
      GO TO 30
53    IC=IC-1
      IC1=IC1-1
      GO TO 55
56    WRITE(6,206)
206   FORMAT('0ALL VARIANCE COMPONENTS ARE ZERO')
      GO TO 22
C      COMPUTE DELTA H AND DET(H)
30    DO 27 I=1,M
      DO 27 J=I,M
27    WM(I,J)=R(I,J)
      LL=1
      LU=0
      DO 28 I=1,IC
      LL=LL+MI(I)
      LU=LU+MI(I+1)
      DO 28 J=LL,LU
28    WM(J,J)=WM(J,J)+1.D0/D(I)
      CALL CDI(M,WM,TOL,DET,MAXMK1)
      DO 70 I=1,M
      DO 70 J=I,M
70    WM(J,I)=WM(I,J)
```

```
      DO 20 I=1,IC
20    DET=DET+DLOG(D(I)**MI(I+1))
C      COMPUTE W + DELTA W
      K2=M+1
      DO 71 J=1,K2
      DO 72 I=1,M
      TEMP=0.
      DO 73 L=1,M
73    TEMP=TEMP+WM(I,L)*R(L,J)
72    WM2(1,I)=TEMP
      DO 71 I=J,K2
      TEMP=0
      DO 74 L=1,M
74    TEMP=TEMP+R(L,I)*WM2(1,L)
      W(J,I)=R(J,I)-TEMP
71    W(I,J)=W(J,I)
      GO TO 6
23    CONTINUE
C         COMPUTE LARGE SAMPLE VARIANCE-COVARIANCE MATRIX
      DO 93 I=1,IC1
      DO 93 J=1,IC1
93    WM(I,J)=0.0D0
      LL=1
      LU=0
      DO 94 IN=1,IC
      LL=LL+MI(IN)
      LU=LU+MI(IN+1)
      DO 86 I=LL,LU
86    WM(IN,IC1)=WM(IN,IC1)+W(I,I)
      LLL=LL
      LLU=LU
      DO 94 JN=IN,IC
      DO 87 I=LL,LU
      DO 87 II=LLL,LLU
87    WM(IN,JN)=WM(IN,JN)+W(I,II)*W(II,I)
      IF (JN.EQ.IC) GO TO 94
      LLL=LLL+MI(JN+1)
      LLU=LLU+MI(JN+2)
94    CONTINUE
      S1=2*SIGMA
      S2=S1*SIGMA
      DO 88 I=1,IC
      DO 89 J=I,IC
89    WM(I,J)=WM(I,J)/2
88    WM(I,IC1)=WM(I,IC1)/S1
      WM(IC1,IC1)=NK/S2
      DO 92 I=1,IC1
      DO 92 J=I,IC1
      W(I,J)=WM(I,J)
      W(J,I)=W(I,J)
92    WM(J,I)=WM(I,J)
      CALL INV2(WM,IC1,0,1,IC1,DD,MAXMK1,0)
      WRITE(6,215)
215   FORMAT('-',4X,'ESTIMATES',10X,'LARGE SAMPLE VARIANCE-COVARIANCE MA
     &TRIX')
      DO 85 I=1,IC
85    WRITE(6,207) D(I), (WM(I,J),J=1,IC1)
      WRITE(6,207) SIGMA, (WM(IC1,J),J=1,IC1)
C         COMPUTE LARGE SAMPLE VARIANCE-COVARIANCE MATRIX
C         TRANSFORMED TO SIGMA SQUARED (I)
      S4=SIGMA*SIGMA
      DO 101 I=1,IC
      DR(I)=-D(I)/SIGMA
101   WM(I,1)=0.0D0
      DO 102 I=1,IC
      DO 102 J=1,IC
```

```
102    WM(I,1)=WM(I,1)+W(I,J)*DR(J)
       T=W(IC1,IC1)
       DO 105 I=1,IC
       T=T+DR(I)*(WM(I,1)+2*W(I,IC1))
       W(I,IC1)=(W(I,IC1)+WM(I,1))/SIGMA
       DO 105 J=I,IC
105    W(I,J)=W(I,J)/S4
       W(IC1,IC1)=T
       DO 104 I=1,IC1
       DO 104 J=I,IC1
104    W(J,I)=W(I,J)
       CALL INV2(W,IC1,0,1,IC1,DD,K4,0)
       WRITE(6,216)
216    FORMAT('-','VARIANCE COMPONENTS',5X,'LARGE SAMPLE VARIANCE-COVARIA
      &NCE MATRIX')
       DO 98 I=1,IC
       SIGI=D(I)*SIGMA
98     WRITE(6,207) SIGI, (W(I,J),J=1,IC1)
       WRITE(6,207) SIGMA, (W(IC1,J),J=1,IC1)
C          TRANSFORM R  ( M1 X M1 ) BACK TO ITS ORIGINAL FORM
       M1=M+1
       M2=M+2
       MMIN=M-1
       DO 65 I=1,M
       DO 65 J=1,K
       W(I,J)=0.0D0
       DO 65 II=1,K
65     W(I,J)=W(I,J)+P(I,M1+II)*R(M1+II,M1+J)
       DO 66 I=1,M
       DO 66 J=1,M
       DO 66 II=1,K
66     R(I,J)=R(I,J)+W(I,II)*R(M1+II,J)
       DO 25 I=1,M
       DO 67 II=1,K
67     R(I,M1)=R(I,M1)+W(I,II)*R(M1+II,M1)
25     R(M1,I)=R(I,M1)
       R(M1,M1)=R(M1,M1)+YY
       CALL INV2(R,K,0,M2,K3,DD,K4,M1)
C          COMPUTE FIXED EFFECTS
       LL=1
       LU=0
       DO 36 I=1,IC
       LL=LL+MI(I)
       LU=LU+MI(I+1)
       DO 36 J=LL,LU
36     R(J,J)=R(J,J)+1.0D0/D(I)
       CALL CDI(M,R,TOL,DET,K4)
       DO 90 I=1,M
       DO 90 J=1,M
90     R(J,I)=R(I,J)
       DO 40 I=1,K
       DO 40 J=1,M
       WM(I,J)=0.0D0
       DO 40 II=1,M
40     WM(I,J)=WM(I,J)+R(M1+I,II)*R(II,J)
       DO 41 I=1,K
       DO 41 J=1,K
       W(I,J)=0.0D0
       DO 41 II=1,M
41     W(I,J)=W(I,J)+WM(I,II)*R(II,M1+J)
       DO 49 I=1,K
       W(I,K1)=0.0D0
       DO 42 II=1,M
42     W(I,K1)=W(I,K1)+WM(I,II)*R(II,M1)
49     W(K1,I)=W(I,K1)
       AB=0.0D0
```

```
      DO 43 J=1,M
      WM2(1,J)=0.0D0
      DO 45 II=1,M
 45   WM2(1,J)=WM2(1,J)+R(M1,II)*R(II,J)
 43   AB=AB+WM2(1,J)*R(J,M1)
      W(K1,K1)=AB
      DO 46 I=1,K
      W(I,K1)=R(M1+I,M1)-W(I,K1)
      W(K1,I)=W(I,K1)
      DO 46 J=1,K
 46   W(I,J)=R(M1+I,M1+J)-W(I,J)
      W(K1,K1)=R(M1,M1)-W(K1,K1)
      GO TO 21
C        COMPUTE FIXED EFFECTS WHEN ALL VARIANCE COMPONENTS ARE ZERO
 22   DO 38 I=1,K
      W(I,K1)=R(I+1,1)
      W(K1,I)=W(I,K1)
      DO 38 J=1,K
 38   W(I,J)=R(I+1,J+1)
      W(K1,K1)=R(1,1)+YY
      CALL INV2(W,K,0,1,K,DD,K4,0)
 21   CALL INV2(W,K,1,1,K,DD,K4,0)
C        COMPUTE VARIANCE-COVARIANCE MATRIX OF THE FIXED EFFECTS
      DO 19 I=1,K
      DO 19 J=1,K
 19   W(I,J)=W(I,J)*SIGMA
      WRITE(6,212)
 212  FORMAT('-',3X,'FIXED EFFECTS',8X,'VARIANCE-COVARIANCE MATRIX')
      DO 47 I=1,K
 47   WRITE(6,207) W(I,K1),(W(I,J),J=1,K)
 207  FORMAT('0',G17.8,3X,6G17.8/' ',20X,6G17.8)
      RETURN
 403  WRITE(6,208)
 208  FORMAT('-','THE LIKELIHOOD IS BEING MINIMIZED, A POSSIBLE CORRECTI
     &ON IS TO TRY DIFFERENT INITIAL VALUES FOR THE GAMMA VECTOR')
      RETURN
      END
      SUBROUTINE INV2(A,N,M,N1,N2,D,M1,NN)

      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(M1,M1)
      N3=1
      IF (NN.NE.0) N3=N1
      L=NN+N+M
      LN=NN+N
      D=1.D0
      DO 1 I=N1,N2
      D=D*A(I,I)
      A(I,I)=1.D0/A(I,I)
      DO 2 J=N3,L
      IF(J-I)3,2,3
 3    A(I,J)=A(I,J)*A(I,I)
 2    CONTINUE
      DO 1 J=N3,LN
      IF(J-I)4,1,4
 4    DO 6 K=N3,L
      IF(K-I)5,6,5
 5    A(J,K)=A(J,K)-A(J,I)*A(I,K)
 6    CONTINUE
      A(J,I)=-A(J,I)*A(I,I)
 1    CONTINUE
      RETURN
      END

      SUBROUTINE CDI(N,A,TOL,D,MAX)
C      INVERSION BY CHOLESKY DECOMPOSITION
```

```
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(MAX,MAX)
C      TRIANGULAR DECOMPOSITION
      DO 2 K=1,N
      K1=K+1
      IF(A(K,K).LE.TOL)GO TO 5
      A(K,K)=DSQRT(A(K,K))
      IF(K.EQ.N)GO TO 3
      DO 1 J=K1,N
1     A(K,J)=A(K,J)/A(K,K)
      DO 2 I=K1,N
      DO 2 J=I,N
2     A(I,J)=A(I,J)-A(K,I)*A(K,J)
C      COMPUTE DETERMINANT OF A
3     D=0.0D0
      DO 4 K=1,N
4     D=D+DLOG(A(K,K))
      D=D+D
C      BACKSOLUTION FOR R INVERSE
      DO 6 K=1,N
      L=N+1-K
      A(L,L)=1.D0/A(L,L)
      IF(L.EQ.1)GO TO 8
      DO 6I=2,L
      I2=L+1-I
      T=0.
      DO 7 M=2,I
      J=L-M+2
7     T=T-A(I2,J)*A(J,L)
6     A(I2,L)=T/A(I2,I2)
8     CONTINUE
C      MULTIPLICATION FOR A INVERSE
      DO 9 I=1,N
      DO 9 J=I,N
      T=0.
      DO 10 K=J,N
10    T=T+A(I,K)*A(J,K)
9     A(I,J)=T
      RETURN
5     WRITE(6,207)
207   FORMAT(' MATRTX Q IS NOT POSITIVE DEFINITE')
      STOP
      END
//LKED.SYSIN DD *
//GO.SYSIN DD *
 OSTLE'S 14 PINE BOARD DATA
   14    1    1    5   20    1   10
    5
(6F3.0,F5.1)
(' ',5F3.0,3X,F3.0,3X,F5.1)
  1              1  7.3
  1              1  8.3
  1              1  7.6
  1              1  8.4
  1              1  8.3
     1           1  5.4
     1           1  7.4
     1           1  7.1
        1        1  8.1
        1        1  6.4
           1     1  7.9
           1     1  9.5
           1     1 10.0
              1  1  7.1
1.0
```

Credit

Portions of the coding have been taken from a program prepared by W. J. Hemmerle and designed after the Hemmerle and Hartley (1973) algorithm.

References

Corbeil, R. R. and S. R. Searle (1974). Restricted maximum likelihood (REML) estimation of variance components in the mixed model. Paper No. BU-538-M in the Biometrics Unit Mimeo Series, Cornell University, Ithaca, New York.

Hemmerle, W. J. and H. O. Hartley (1973). Computing maximum likelihood estimates for the mixed A.O.V. model using the W-transformation. Technometrics 15:819-832.